

Navigating Provenance Information for Distributed Healthcare Management

Vikas Deora, Arnaud Contes, Omer F. Rana, Shrija Rajbhandari, and Ian Wootten
School of Computer Science, Cardiff University
Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, UK
v.deora@cs.cardiff.ac.uk

Kifor Tamas and Laszlo Z. Varga
Computer and Automation Research Institute
Kende u. 13-17, 1111 Budapest, Hungary
tamas.kifor@sztaki.hu

Abstract

Provenance information provides a useful basis to verify whether a particular application behavior has been adhered to. This is particularly useful to evaluate the basis for a particular outcome, as a result of a process, and to verify if the process involved in making the decision conforms to some pre-defined set of rules. This is significant in a healthcare scenario, where it is necessary to demonstrate that patient data has been processed in a particular way. Understanding how provenance information may be recorded, stored, and subsequently analyzed by a decision maker is therefore significant in a service oriented architecture, which involves the use of third party services over which the decision maker does not have control. The aggregation of data from multiple sources of patient information plays an important part in subsequent treatments that are proposed for a patient. A tool to navigate through and analyze such provenance information is proposed, based on the use of a portal framework that allows different views on provenance information to co-exist. The portal enables users to add custom portlets enabling application specific views that would facilitate particular decision making.

1 Introduction

Service oriented architecture (SOA) and standardized electronic healthcare record exchange techniques [1] are beginning to make it possible to combine information about different stages of therapy received by a single patient from different healthcare providers. The SOA approach achieves this by treating each data source as an independent service. However, this process only allows integration and sharing of business logic at a functional level and does not yet ad-

dress the presentation of such information to an end user. This makes it imperative for users to handle the presentation independently.

Healthcare data is often distributed among several heterogeneous and autonomous information systems (actors), under different healthcare authorities, like general practitioners, hospital departments, etc. Each actor operates independently and defines their processes and data representation. As a patient undergoes different stages of therapy, a particular treatment's history is generated, we refer to this as "provenance" of that treatment [2]. Such provenance information plays an important part in subsequent treatment of a patient.

For example, consider a patient visiting an eye clinic to report loss of vision. This could be a result of treatments undertaken in the past. The doctor would like to trace any past medical history that could have caused such a condition, involving analysis on data at remote locations. This process would ideally require the following:

- The interface to support interaction with various services that use standard interfaces such as web browsers.
- A Single Sign On (SSO) based authentication mechanism.
- A presentation interface that can be customized for different categories of application users.

Current SOA based frameworks are not able to handle such requirements. For example, in the above scenario a doctor would have to understand the protocol to communicate with different services, negotiate security requirements, retrieve data and perform analysis on such data independently. We propose a framework based on the use of a portal and

portlets that follows the architectural constraints of SOA (such as Web services), but one that also provides presentation interfaces for applications such as an Electronic Health-Care Record (EHCR) system, we discuss this application further in Section 2. We also demonstrate how provenance of a patient's treatment can be identified and navigated using developed portlets (discussed in Section 4).

1.1 Related Work

Some work has been performed previously using portal frameworks in healthcare [3, 4] for dissemination of healthcare related information. There are other projects that are using portal and portlets to access distributed resources, specially in the context of supporting collaborative simulation [5], process monitoring and execution [6], University support system [7], etc. Our work differs mainly for two reasons: first, in exploiting provenance information captured from resource execution, while existing work involves using portal framework to facilitate the collection and execution of distributed resources. Second, our work provides auditing to verify if a particular process was executed as expected, while other approaches are concerned with composing a process to achieve such results.

Work exists in area of process visualization, such as Taverna [8], Triana [9], DAGMan [10], etc. The aim of such projects is to help a user compose and execute a process. However, our work assumes a process already exists and to use the provenance of such process to provide decision support. Existing approaches also use specific process description languages (e.g. [11]) and a centralized engine for process execution (e.g. [12]). However, no such assumption is made by us; the provenance information only contains information about interactions between two actors and the relationships between interactions. The provenance information is asserted in a standard format [13] by the actors involved in interactions.

The rest of the paper is organized as follows. Section 2 gives background to the EHCR system. Section 3 describes the portal and the relation to SOA and EHCR. Section 4 describes the developed portlets. Section 5 concludes and presents future work.

2 Electronic HealthCare Record System

The EHCR architecture provides the structure to build partial/full patient healthcare record from multiple heterogeneous database systems. It uses the ENV13606 standard [1], which defines the messages, the retrievable objects, the healthcare services and the distribution rules. Although the EHCR architecture defines how to exchange data [14], the linking of process information which generated the data is

not discussed in EHCR. The provenance architecture [13] helps to document the way a data item was created. The portal and portlets (discussed in Section 3) provides the necessary tools to navigate and link together process documentation.

The ENV13606 standard has three types of messages: (1) a request message that contains a reason for the request, (2) a notification message that contains the type and comment of the notification, and (3) a provide message that contains privacy protection rules. These three types of messages also contains: the identification of the message, the issue date and time of the message, the EHCR source/destination service, urgency of the message, patient matching information (the subject of the message) and message receipt acknowledgement request. Further discussion on EHCR system is available at [14].

3 Portal and Service Oriented Architecture

Current SOA provided a mechanism to integrate and share business logic at functional level; however it does not yet address the presentation issues for the consumers of a service, thus making it imperative for consumers to handle the presentation independently. The presentation layer provided by a portal framework is based on Web based interfaces, thus making it easy for consumers to use existing browsers.

Two main portal standards exist today: Java Specification Request (JSR) 168 [15] and Web Service for Remote Portlets (WSRP) [16]. Not all portal vendors support these two standards (e.g. [17]) and some support their own modified versions of these standards [18]. In order to meet the constraints of SOA, it is imperative that a framework that supports both WSRP and JSR 168 is selected.

JSR 168 enables interoperability between portlets and portals, by defining an API for aggregation, personalization, presentation and security. JSR 168 specifies how a portlet should be implemented and deployed, but not how communication between two portlets should take place. WSRP on the other hand simplifies the effort required for integrating portlets/applications. The API defined by WSRP specification uses the Web Services Description Language (WSDL), utilizing existing specifications such as WS-Policy. WSRP specification defines the communication protocol required for consumers; in this case a portal to find and integrate geographically distributed services made available by portlet producers. In Figure 1 each provider portlets in portal containers (Hospital A and C) generates fragments of mark-up which the consumer portal (Hospital A) ultimately pieces together to create a complete page that is presented to the user.

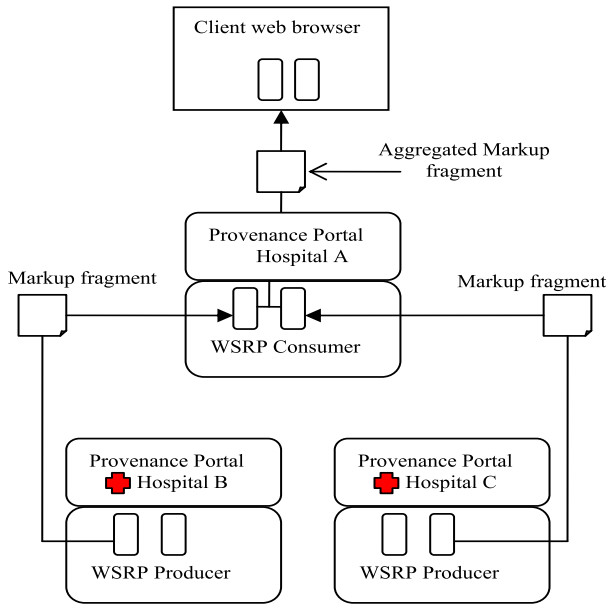


Figure 1. Producer and consumer model of WSRP

3.1 Portal and Electronic HealthCare Record System

Healthcare activities depend not only on the actual medical process, but also on the organization carrying out the activity, the legal regulations, cultural aspects, preferences of health professional groups, etc. Such information systems can often consist of deeply specialized products covering various aspects of hospital information management. As patients may have treatments in geographically dispersed locations over time, this process results in collection of heterogeneous data about a patient in different organizations. The aim of the portal is to provide access to this EHCR information in a way that allows healthcare users from different organizations to personalize their portal with services they would like to use.

Our portal is based on the eXo portal framework [19]. Some of the common features that are provided by eXo and most other portal frameworks (such as Liferay, Gridsphere, etc) and the relationship to EHCR is:

- **Personalization:** allows the presentation and information contents of portal pages to be customized for a specific user. In eXo this is achieved by gathering and displaying all the contents relevant to a particular user profile (e.g. doctors, hospital managers, etc).
- **Aggregation:** allows portal content from various disparate sources to be combined and presented to a user

in a single response. This is achieved by invoking portlets from distributed portal containers to produce content that becomes a fragment of a document that is finally presented to the user. Aggregation provides a user with access to content that is outside their current domain. Aggregation is achieved in our framework using the WSRP (as shown in Figure 1). This feature is important for users, as it is required by each healthcare organization to maintain their individual set of portlets within a local portal container. However, since doctors can belong to one or more hospitals, this requires access to portlets from remote hospitals portlet containers.

- **Single Sign-On (SSO):** provides an authentication mechanism by which the user can access external content without having to authenticate again. As the portal requires access to many distributed data stores for retrieval of patient data, SSO provides a secure and unrestricted access to approved users.

Within our portal, SSO is achieved by integration of eXo portal and Community Authorization Service (CAS). CAS framework [20] provides a public key authentication and delegation mechanisms that supports single sign-on. CAS allows resource owners (i.e. actors) to grant access to blocks of resources to a community (e.g., surgeon) as a whole, and let the community itself manage fine-grained access control within that framework.

Our portal in this scenario does not render the default eXo portal login page but delegates the authentication to CAS login page. In the CAS login page, a check is performed to verify if CAS has already granted a certificate and if it is a valid one. If valid, the portal redirects the user to his/her default portal page. Once authenticated, a user's certificate is set as a session object to be made available to all active portlets as credential to log into any external resources. However, if authentication fails, the login module directs the user to a CAS form that allows a user to authenticate using username/password pair. With a valid username/password pair a certificate is set at the client end for the user.

3.2 Portal-EHCR Architecture

An architecture of using the portal framework with EHCR is shown in Figure 2. The architecture consists of provenance-aware actors, EHCR, Provenance Store and portal and portlets. Each healthcare organization in our system must include these four components.

A set of provenance-aware actors involved in a process generate data about the execution. The data produced is composed of a set of *p-assertions*. Such set of *p-assertions*

provide the description of the physical process [13]. A p-assertion can be used to record one of the following events: an *interaction* between two actors, a *relation* between two events, or the state of an actor at a particular moment. In our system, interaction and relationship p-assertion are presently used (these are discussed in Section 4.2). As seen in Figure 2 these p-assertions are stored within a *Provenance Store* (PS) using EHCR application.

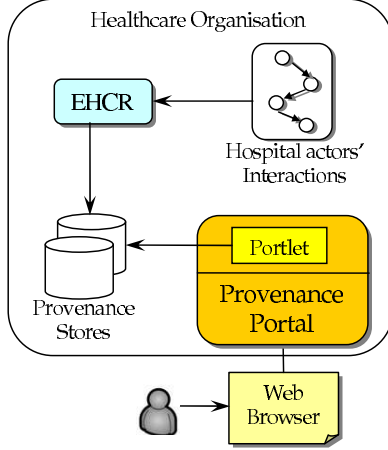


Figure 2. Portal-EHCR architecture

The portal and portlets provide users with a set of tools to navigate through and analyze a set of p-assertions that represent an executed process. Interaction with a portal is made available using a Web client (browser). On receiving a user request to re-construct a patient history, the portlet interacts with local and external PSs to retrieve all the p-assertions related to a particular process execution.

4 Portlets

As stated previously in section 3.2, two kinds of p-assertions (interaction and relationship) exist in our system. In this section, we describe the three portlets that together provides support for navigation, visualization and analysis of such p-assertions.

4.1 Textual Navigation Portlet

With the textual navigation portlet as shown in Figure 3, a user is able to send personalized queries to the PSs. This portlet retrieves the results from multiple PSs and displays them in an XML form, a tree and as an XSL Transformations (XSLT) generated markup. The XML and tree view are standard ways to display XML data, thus we concentrate our discussion on the XSLT transformation.

The history of a patient may contain data over a long period of time. During this period, the content and the format of the electronic data collected may have changed, due to technology improvements, patient's illness, discovering of new diseases, etc. Therefore, the navigation portlet must be flexible enough to provide customized display of query results. This is achieved by allowing XSLT transformation. This process provides a filtering layer, which is inserted between the patient's data and the application user. The purpose of this layer is to extract the accurate pieces of information among all available information according to the current context. An example of this is shown in figure 3, where information on actors involved in a process execution is extracted and displayed in a tabular form.

4.2 Visualization Portlet

Our visualization portlet allows p-assertions retrieved for a given patient, using the navigation tool, to be visualized as a process graph. The visualization portlet displays two process graphs: interaction graph and relationship graph that are based on interaction and relationship p-assertions respectively. Each of these is discussed below:

- **Interaction graph:** In the context of SOAs, interactions consist of the messages exchanged between actors. By capturing all the interactions that take place between actors involved in the computation of some data, one can replay an execution, analyze it, verify its validity or compare it with another execution. A crucial element of an interaction p-assertion is information to identify a message uniquely. Such information allows us to establish a flow of data between actors. Indeed, let us consider two interaction p-assertions: actor A making an assertion α_A that it sent actor B a message with identity i , and actor B making an assertion α_B that it received from A a message with the same identity i . Such a pair of interaction p-assertions α_A, α_B is said to be matching; it identifies a flow of data from actor A to B.

Figure 4 displays a re-constructed patient history from distributed PSs using interaction p-assertions. In this case, the actors are represented as *boxes* and the *edges* represents the interaction between the actors. Multiple edges between two actors represent multiple interactions. As can be seen from Figure 4, five actors are involved in the process. The Radiology unit, Orthopedic Unit, Rheumatology Unit, Ultrasound unit, and Vision unit. Figure 4 displays all interaction between actors as part of past treatments received by a patient.

- **Relationship graph:** While matching interaction p-assertions denote a flow of data between actors, relationships explain how data flows inside actors. Rela-

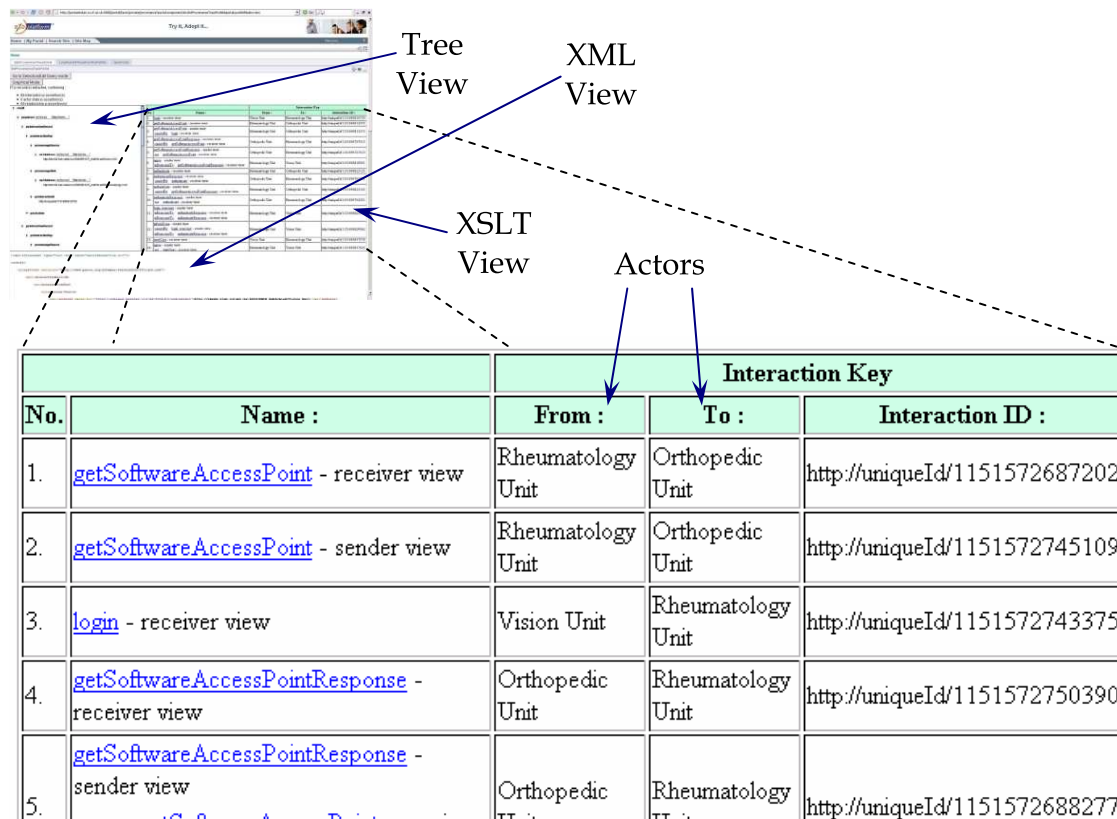


Figure 3. Textual navigation portlet

tionship p-assertions are directional since they explain how some data was computed from other data. Figure 4 displays the relationship graph that illustrates the relationship between the interactions that took place as part of a process. In this case, a interaction is represented as *boxes* and the *edges* represents the relationship between the interactions. The relationship graph helps a user understand "Why" and "How" a interaction happened. Relationship view also allows a user analyze any critical interactions. For example, in Figure 4 a interaction (joint replacement decision(4)) is caused due to other interactions happening (e.g. orthopedic report(1), radiology test result(2), and rheumatologist report(3)). Further investigation on such processes can be performed to trace how a treatment decision was made. Such analysis also helps in auditing; for example, to verify if all necessary interactions were performed before a decision was made.

Interaction p-assertions denote data flows between actors, whereas relationship p-assertions denote private ones. Such data flows are core elements to reconstitute functional data dependencies in execution. From a specific data item,

the data flow directed acyclic graph indicates where and how the data item is being used; vice-versa, following relationships in reverse helps us identify how a data item was produced.

4.3 Analysis Portlet

The analysis portlet provides the capabilities to analyze the retrieved graphs. The analysis portlet is based on the Java Expert System Shell (JESS), a java rule engine. JESS uses an enhanced version of the Rete algorithm to process rules. Rete is an efficient mechanism for solving the difficult many-to-many matching problem (see for example [21]). The Rete algorithm expects two different type of input, (1) a set of *rules* which represent the logic of the computation (also called *production rules*) and (2) a set of facts which represent the data to be analyzed (also called *working memory*).

On top of providing some reasoning framework within the project, the analysis tool is also used to detect possible conflicts in the p-assertions recorded. The nature of detected conflicts is large and various, from detecting a difference between the data submitted by the sender and

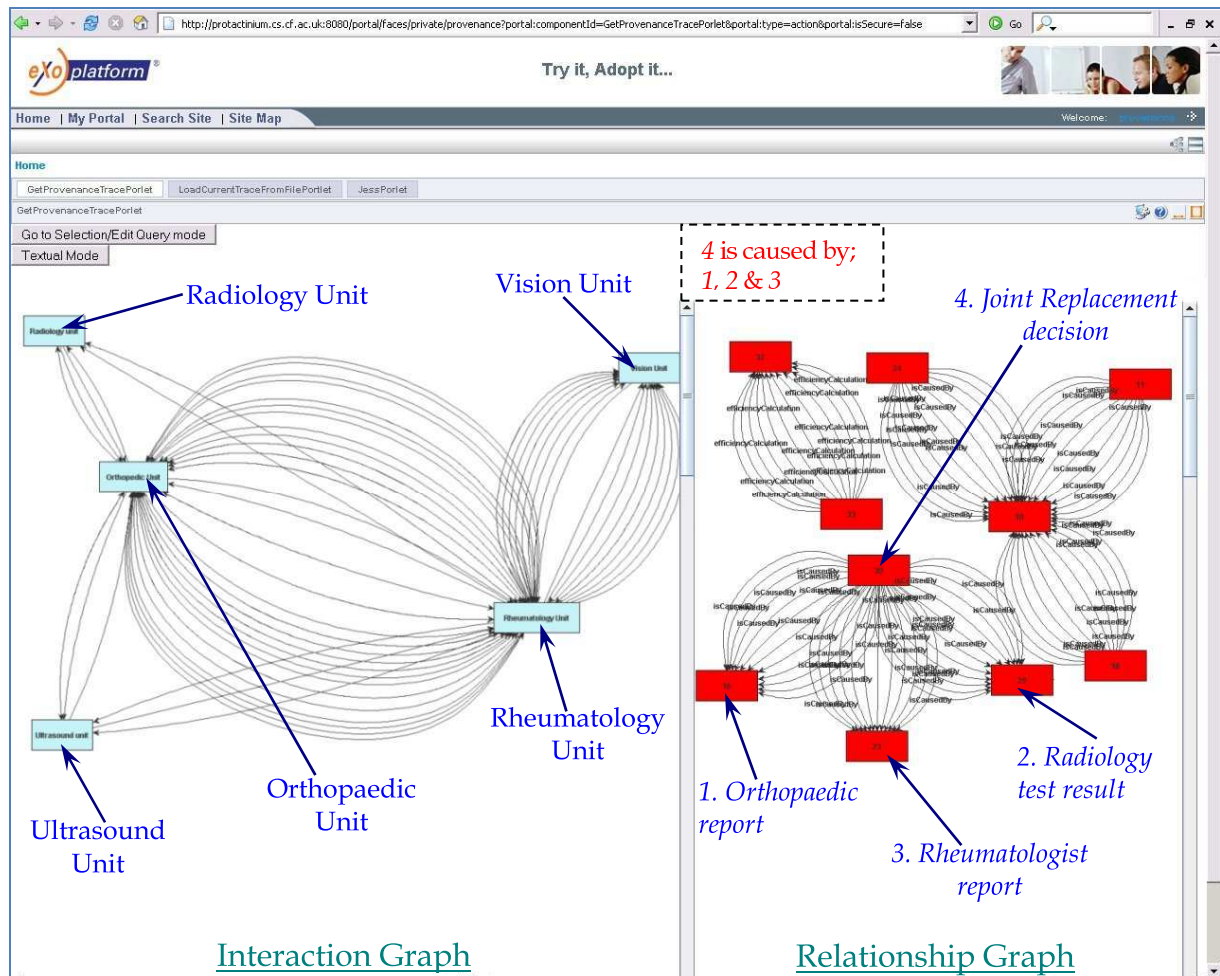


Figure 4. Interaction graph and relationship graph

by the receiver of a same interaction (e.g., if the result of a blood test as reported by a laboratory and the result available at a particular hospital differ) or the detection of unexpected behavior while the execution of a process (for instance, an abnormal long duration between the extraction of an organ and the scheduled transplant operation).

The three portlets: textual, visualization and analysis provide decision support tools for doctors. This is especially important when the treatment is directly affected by past medical history. The visualization portlet allows a complete patient history to be re-constructed, allowing doctors to trace how important decisions were made. The analysis portlet provides users with audit support, and help detect anomalies in past treatments. Together, these set of portlets provide enough content to allow doctors to make informed choices.

5 Conclusion and Future Work

We demonstrated work on portal and portlets to provide a set of tools that allow users to re-construct and visualize a patient's medical history. We have also demonstrated how the above can be performed in a secure way using single sign-on (SSO). Using WSRP, our portlets can be utilized not just by partner healthcare applications but also other external applications. This allows re-usability of developed portlets. All portlets are JSR 168 compliant, and deployable under any compliant portlet containers. Portal and portlets are still in early stages of research and many questions still need to be answered, such as automated discovery and selection of portlets, an efficient way to handle inter-portlet communication and security. These issues will form the topic of our further investigation.

Our work provides a complimentary approach to many existing process composition tools such as [9]. We would

like to further evaluate our work based on use of our tools in such projects. Also, current process reconstruction portlet is not modeled to display a large amount of provenance information. Design challenges in visualizing complex processes involving large number of services is under study.

Acknowledgments

We would like to thank the other Provenance project partners for their input: IBM United Kingdom, University of Southampton, Deutsches Zentrum für Luft- und Raumfahrt, Universitat Politècnica de Catalunya.

References

- [1] ENV13606: Health informatics electronic healthcare record communication part 1: Extended architecture and domain model. Final Draft prENV13606-1 (1999)
- [2] Groth, P., Luck, M., Moreau, L.: Formalising a protocol for recording provenance in grids. In: In Proceedings of The UK OST e-Science second All Hands Meeting 2004 (AHM'04). (2004)
- [3] Shepherd, M., Zitner, D., Watters, C.: Medical portals: Web-based access to medical information. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. (2000) 1 – 10
- [4] Murray, M.: An investigation of specifications for migrating to a web portal framework for the dissemination of health information within a public health network. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences. (2002) 1917 – 1925
- [5] Lin, M., Walker, D.W., Chen, Y., Jones, J.W.: A grid-based problem solving environment for gecem. In: IEEE International Symposium on Cluster Computing and the Grid. Volume 2. (2005) 686 – 693
- [6] Csaba, N., Gabor, D., Robert, L., Peter, K.: The p-grade grid portal. In: Proceedings of the International Conference on Computational Science and Its Applications. (2004)
- [7] Fengchun, Z., Aihua, W., Yanbing, J.: A framework to develop a university information portal. In: Proceedings of the International Conference on Information Acquisition. (2004) 506 – 509
- [8] Oinn, T.M., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, R.M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* **20**(17) (2004) 3045–3054
- [9] Taylor, I., Wang, I., Shields, M.S., Majithia, S.: Distributed computing with triana on the grid. *Concurrency - Practice and Experience* **17**(9) (2005) 1197–1214
- [10] Condor: Directed acyclic graph manager (dagman). <http://www.cs.wisc.edu/condor/dagman/>, University of Wisconsin-Madison (2006)
- [11] Alves, A., Arkin, A., Askary, S., Bloch, B., Curbera, F., Goland, Y., Kartha, N., Liu, C.K., Knig, D., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A.: Web services business process execution language. <http://www.oasis-open.org/committees/download.php/18714/wsbpel-specification-draft-May17.htm> (2006)
- [12] ActiveBPEL. <http://www.activebpel.org/> (2006)
- [13] Groth, P., Jiang, S., Miles, S., Munroe, S., Tan, V., Tsasakou, S., Moreau, L.: An architecture for provenance systems. Technical report, University of Southampton (2006)
- [14] Kifor, T., Laszlo, V., Alvarez, S., Vazquez-Salceda, J., Willmott, S.: Privacy issues of provenance in electronic healthcare record systems. In: In Proceedings of 1st Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE 2006), 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006). (2006)
- [15] Abdelnur, A., Chien, E., Hepper, S.: Jsr-000168 portlet specification. URL (October)
- [16] Thompson, R.: Web services for remote portlets specification v2.0 (2006)
- [17] GridSphere. <http://www.gridsphere.org> (2006)
- [18] JBoss. <http://www.jboss.org/products/jbossportal> (2006)
- [19] Mestrallet, B., Nguyen, T., Azarenkov, G., Moron, F., Revenant, B.: exo platform v2, portal, jcr, ecm, groupware and business intelligence. url (2005)
- [20] Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: The community authorization service: Status and future. In: n Proceedings of Computing in High Energy Physics 03 (CHEP '03). (2003)
- [21] C. L. Forgy: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence* **19** (1982) 17–37